Create a Structure Bank Account with following details:

Bank Name, IFSC code, Account Number, Account Holder name, Age, Gender,

DOB, Address, City, Type of Account, Balance, Pan Card number, Aadhar number.

1. Create Account

2. Update Account Details

a. Update name of account holder

b. Update address of account holder

c. Update DOB of account holder

3. Deposit

4. Withdraw

5. Funds Transfer

6. Search details of account holder

a. Search by account number

b. Search by name

c. Search by Type of Account

7. Balance Enquiry

PROGRAM:

#include <stdio.h>

#include <string.h>

// Structure to represent a Bank Account

struct BankAccount {

char bank\_name[50];

char ifsc\_code[12];

char account\_number[20];

char account\_holder\_name[50];

int age;

char gender;

char dob[15];

char address[100];

char city[50];

char account\_type[20];

float balance;

char pan\_card\_number[20];

char aadhar\_number[20];

};

struct BankAccount createAccount() {

struct BankAccount newAccount;

printf("Enter Bank Name: ");

scanf("%s", newAccount.bank\_name);

printf("Enter IFSC Code: ");

scanf("%s", newAccount.ifsc\_code);

printf("Enter Account Number: ");

scanf("%s", newAccount.account\_number);

printf("Enter Account Holder Name: ");

scanf("%s", newAccount.account\_holder\_name);

printf("Enter Age: ");

scanf("%d", &newAccount.age);

printf("Enter Gender (M/F): ");

scanf(" %c", &newAccount.gender);

printf("Enter Date of Birth (DD-MM-YYYY): ");

scanf("%s", newAccount.dob);

printf("Enter Address: ");

scanf("%s", newAccount.address);

printf("Enter City: ");

scanf("%s", newAccount.city);

printf("Enter Type of Account: ");

scanf("%s", newAccount.account\_type);

printf("Enter Balance: ");

scanf("%f", &newAccount.balance);

printf("Enter Pan Card Number: ");

scanf("%s", newAccount.pan\_card\_number);

printf("Enter Aadhar Number: ");

scanf("%s", newAccount.aadhar\_number);

printf("Account created successfully!\n");

// Display the entered values

printf("New Account Details:\n");

printf("Bank Name: %s\n", newAccount.bank\_name);

printf("IFSC Code: %s\n", newAccount.ifsc\_code);

printf("Account Number: %s\n", newAccount.account\_number);

printf("Account Holder Name: %s\n", newAccount.account\_holder\_name);

printf("Age: %d\n", newAccount.age);

printf("Gender: %c\n", newAccount.gender);

printf("DOB: %s\n", newAccount.dob);

printf("Address: %s\n", newAccount.address);

printf("City: %s\n", newAccount.city);

printf("Type of Account: %s\n", newAccount.account\_type);

printf("Balance: %.2f\n", newAccount.balance);

printf("Pan Card Number: %s\n", newAccount.pan\_card\_number);

printf("Aadhar Number: %s\n", newAccount.aadhar\_number);

printf("Account created successfully!\n");

return newAccount;

}

// Function to update account details

void updateAccount(struct BankAccount \*account, char option) {

switch(option) {

case 'a':

printf("Enter new name of account holder: ");

scanf("%s", account->account\_holder\_name);

break;

case 'b':

printf("Enter new address of account holder: ");

scanf("%s", account->address);

break;

case 'c':

printf("Enter new DOB of account holder (DD-MM-YYYY): ");

scanf("%s", account->dob);

break;

default:

printf("Invalid option\n");

return;

}

// Display the updated values

printf("Updated Account Details:\n");

printf("Account Holder Name: %s\n", account->account\_holder\_name);

printf("Address: %s\n", account->address);

printf("DOB: %s\n", account->dob);

}

// Function to deposit money

void deposit(struct BankAccount \*account, float amount) {

account->balance += amount;

// Display updated balance

printf("Amount deposited successfully!\n");

printf("Updated Balance: %.2f\n", account->balance);

}

void withdraw(struct BankAccount \*account, float amount) {

if (amount <= account->balance) {

account->balance -= amount;

// Display updated balance

printf("Amount withdrawn successfully!\n");

printf("Updated Balance: %.2f\n", account->balance);

} else {

printf("Insufficient funds!\n");

}

}

// Function to transfer funds

void transferFunds(struct BankAccount \*sender, struct BankAccount \*receiver, float amount) {

if (amount <= sender->balance) {

sender->balance -= amount;

receiver->balance += amount;

// Display updated balances

printf("Funds transferred successfully!\n");

printf("Updated Balance (Sender): %.2f\n", sender->balance);

printf("Updated Balance (Receiver): %.2f\n", receiver->balance);

} else {

printf("Insufficient funds!\n");

}

}

// Function to search account details

void searchDetails(struct BankAccount accounts[], int numAccounts, char option, char keyword[]) {

switch(option) {

case 'a':

for (int i = 0; i < numAccounts; i++) {

if (strcmp(accounts[i].account\_number, keyword) == 0) {

// Display account details

printf("Account Holder Name: %s\n", accounts[i].account\_holder\_name);

printf("Balance: %.2f\n", accounts[i].balance);

return;

}

}

printf("Account not found\n");

break;

case 'b':

for (int i = 0; i < numAccounts; i++) {

if (strcmp(accounts[i].account\_holder\_name, keyword) == 0) {

// Display account details

printf("Account Number: %s\n", accounts[i].account\_number);

printf("Balance: %.2f\n", accounts[i].balance);

return;

}

}

printf("Account not found\n");

break;

case 'c':

for (int i = 0; i < numAccounts; i++) {

if (strcmp(accounts[i].account\_type, keyword) == 0) {

// Display account details

printf("Account Holder Name: %s\n", accounts[i].account\_holder\_name);

printf("Account Number: %s\n", accounts[i].account\_number);

printf("Balance: %.2f\n", accounts[i].balance);

return;

}

}

printf("Account not found\n");

break;

default:

printf("Invalid option\n");

}

}

// Function to check account balance

float balanceEnquiry(struct BankAccount \*account) {

return account->balance;

}

int main() {

struct BankAccount accounts[10];

int numAccounts = 0;

char option;

do {

printf("\nMenu:\n");

printf("1. Create Account\n");

printf("2. Update Account Details\n");

printf(" a. Update name of account holder\n");

printf(" b. Update address of account holder\n");

printf(" c. Update DOB of account holder\n");

printf("3. Deposit\n");

printf("4. Withdraw\n");

printf("5. Funds Transfer\n");

printf("6. Search details of account holder\n");

printf(" a. Search by account number\n");

printf(" b. Search by name\n");

printf(" c. Search by Type of Account\n");

printf("7. Balance Enquiry\n");

printf("0. Exit\n");

printf("Enter your choice: ");

scanf(" %c", &option);

switch (option) {

case '1':

// Creating accounts

accounts[numAccounts++] = createAccount();

break;

case '2':

// Updating account details

printf("Enter option (a, b, or c): ");

scanf(" %c", &option);

updateAccount(&accounts[0], option);

break;

case '3':

// Deposit

printf("Enter amount to deposit: ");

float depositAmount;

scanf("%f", &depositAmount);

deposit(&accounts[0], depositAmount);

break;

case '4':

// Withdraw

printf("Enter amount to withdraw: ");

float withdrawAmount;

scanf("%f", &withdrawAmount);

withdraw(&accounts[0], withdrawAmount);

break;

case '5':

// Transfer funds

printf("Enter amount to transfer: ");

float transferAmount;

scanf("%f", &transferAmount);

transferFunds(&accounts[0], &accounts[1], transferAmount);

break;

case '6':

// Search account details

printf("Enter option (a, b, or c): ");

scanf(" %c", &option);

char keyword[50];

printf("Enter search keyword: ");

scanf("%s", keyword);

searchDetails(accounts, numAccounts, option, keyword);

break;

case '7':

// Balance Enquiry

printf("Balance: %.2f\n", balanceEnquiry(&accounts[0]));

break;

case '0':

printf("Exiting the program\n");

break;

default:

printf("Invalid option. Please try again.\n");

}

} while (option != '0');

return 0;

}